Cryptanalysis of Two Variants of PCBC Mode when used for Message Integrity

Chris J. Mitchell

Information Security Group, Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
http://www.isg.rhul.ac.uk/~cjm

Abstract. The PCBC block cipher mode of operation has many variants, of which one, due to Meyer and Matyas, dates back over 20 years. Whilst a particularly simple variant of PCBC has long been known to be very weak when used for data integrity protection, the Meyer-Matyas variant has not previously been attacked. In this paper we cryptanalyse this mode, and show that it possesses a serious weakness when used for data integrity protection. Specifically, we show how to construct an existential forgery using only a single known ciphertext message and a modest amount of known plaintext (this could be as little as three plaintext blocks). We also describe a ciphertext-only existential forgery attack against another, recently proposed, PCBC-variant called M-PCBC.

1 Introduction

Traditionally, the recommended way to use a block cipher to provide both integrity and confidentiality protection for a message has been to compute a CBC-MAC and also encrypt the data, using two distinct secret keys. This approach is rather unattractive for some applications because it requires each block of data to be processed twice. This observation has given rise to a number of proposals for combining encryption and integrity protection, including a particular way of using the so called PCBC mode (see, for example, Section 9.6 of [1]).

At the same time, two major problems have recently been identified which have highlighted the need for better-defined integrity and confidentiality modes. Firstly, issues have been identified with certain combinations of encryption and use of a CBC-MAC — see, for example, Bellare, Kohno and Namprempre [2]. That is, it is vital to define precisely how the two operations are combined, including the order of the computations; otherwise there is a danger of possible compromise of the data. Secondly, even where integrity is not explicitly required by the application, if integrity is not provided then in some cases padding oracle attacks may be used to compromise secret data (see, for example, [3–6]).

This has given rise to a number of proposals for well-defined authenticated-encryption modes, including OCB [7], EAX [8] and CCM [9, 10]. These techniques are also the subject of ongoing international standardisation efforts — the first working draft of what is intended to become ISO/IEC 19772 on authenticated encryption was published early in 2004 [11] (see also Dent and Mitchell,

[12]). OCB is to some extent analogous to the use of PCBC to provide integrity, in that it involves only a single pass through the data; unlike PCBC, it also possesses a proof of security.

In this paper we examine two different PCBC variants. We first examine a variant, identified below as PCBC+, which was one of the first ever proposals for a block cipher mode of operation designed to provide both integrity and confidentiality protection. We show that this variant is subject to a known plaintext attack, and hence does not provide adequate integrity protection. We go on to examine a newly proposed variant called M-PCBC [13] which is also claimed to provide both integrity and confidentiality protection when used appropriately. Unfortunately, as shown below, this claim is not correct.

It is important to note that the term PCBC is used by different authors to means slightly different things, and hence we first describe what we mean by PCBC. This is followed by analyses of PCBC+ and M-PCBC.

2 PCBC mode and variants

The precise origin and definition of PCBC, is unclear. In fact, the acronym PCBC has been used to mean two different things, and we define them both.

2.1 The more general definition

Section 9.6 of [1] (Example 9.91) defines the *Plaintext-Ciphertext Block Chaining* mode of operation as follows (note that a special case of this definition goes back at least to 1982, since it is contained in the third and subsequent printings of Meyer and Matyas's 1982 book [14]).

First suppose that the data is to be protected using an n-bit block cipher, i.e. a block cipher operating on plaintext and ciphertext blocks of n bits. We write $e_K(P)$ for the result of block cipher encrypting n-bit block P using the secret key K, and $d_K(C)$ for the result of block cipher decrypting the n-bit block C using the key K. Suppose the plaintext to be protected is divided into a sequence of n-bit blocks (if necessary, first having been padded): $P_1, P_2, \ldots, P_t$.

Then, if the n-bit Initialisation Vector (IV) is S, the PCBC encryption of the plaintext $P_1, P_2, \ldots, P_t$ is defined as:

$$C_i = e_K(P_i \oplus G_{i-1}),\ \ 1 \leq i \leq t,$$

where $G_0 = S$, $G_i = g(P_i, C_i)$, $1 \leq i \leq t$, and g is a simple function that maps a pair of n-bit blocks to a single n-bit block.

Menezes, Van Oorschot and Vanstone [1] make two remarks regarding the choice of g. Firstly they suggest the use of $g(P, C) = P + C \bmod 2^n$, where the n-bit blocks P and C are treated as integers by regarding them as binary representations, and the modulo 2^n sum is converted back to an n-bit block by taking the binary representation (left-padded as necessary with zeros). Secondly they suggest that g should not be equal to the bit-wise exclusive-or of the two

inputs when the mode is to be used to protect the integrity of data (a precaution which they point out is necessary to avoid a known-plaintext attack).

The first choice for g listed above is the technique that is also described by Meyer and Matyas [14]. For convenience we call this mode PCBC+. It is this mode that we consider in detail in this paper. Despite the fact that it has been included in two well known books on cryptography, there would not appear to be any literature at all discussing the security of PCBC+.

We should, at this point, explain how PCBC+ mode (or any other variant of PCBC) can be used to provide both encryption and integrity-protection. The idea is very simple. First divide the data to be encrypted into a sequence of n-bit blocks, padding as necessary. Then append an additional n-bit block to the end of the message, where this block can be predicted by the decrypter (e.g. a fixed block). When the message is decrypted, a check is made that the final block is the expected value and, if it is, then the message is deemed authentic.

Before proceeding observe that this general approach possesses an intrinsic weakness. That is, suppose that a fixed final block (the *terminator block*) is used to detect message manipulations (as above). Then an attacker might be able to persuade the legitimate originator of protected messages to encrypt a message which contains the fixed terminator block somewhere in the middle of the message. The attacker will then be able to delete all ciphertext blocks following the encrypted terminator block, and such a change will not be detectable. Despite this weakness, using an appropriate encryption mode combined with a method for adding verifiable redundancy to a message is still used for message integrity protection — e.g. in Kerberos (see, for example, [12]). As far as this paper is concerned we note that such an attack could be prevented by ensuring that the legitimate encrypter refuses to encrypt any plaintext message containing the terminator block. We further note that such an attack requires chosen plaintext, and the attacks we demonstrate later in this paper require only either a limited amount of known plaintext, or just known ciphertext.

2.2 The more specific definition

PCBC is also sometimes defined [15, 16] to mean *Plaintext Cipher Block Chaining*. In this case PCBC mode is a special case of PCBC as defined above, i.e. where $g(P, C) = P \oplus C$ and where $\oplus$ represents the bit-wise exclusive-or of the blocks P and C. To avoid any confusion we refer to the version of PCBC defined in Menezes et al. [1], and Section 2.1 above, as G-PCBC (for *Generalised PCBC*), and the specific case where g is equal to exclusive-or simply as PCBC.

PCBC in this more specific sense was used in Kerberos version 4 [16] to provide encryption and integrity protection. This was achieved by the means described above, i.e. by checking the final decrypted block of the message.

It is important to observe that PCBC is precisely the version of G-PCBC that Menezes, van Oorschot and Vanstone [1] state should not be used to protect the integrity of data! The weakness of PCBC for use as an integrity-protection mode was first pointed out by Kohl [15]. As is simple to verify, Kohl pointed out that if two of the ciphertext blocks of a PCBC-encrypted message are interchanged,

then this does not affect the decryption of the final block, i.e. it is extremely simple to make undetectable changes to messages. Note that this is actually a stronger attack than is implied by [1] who refer only to the danger of known-plaintext attacks. Finally note that yet another variant of PCBC was proposed by Gligor and Donescu [17]; however, this scheme, known as iaPCBC, was shown to possess serious vulnerabilities by Ferguson et al. [18].

3 IV management strategy for G-PCBC

In the definition of G-PCBC in Section 2.1, the encrypter and decrypter are required to have access to the same n-bit IV, S. However, it is not completely clear from the discussions in [1, 14] how this is meant to be achieved, although it is clear that S should be different for each message. One possible approach is for the sender to choose S and send it in plaintext with the encrypted message. In such a case S might either be chosen at random or generated by a simple counter. In the latter case S will potentially be predictable to an intercepting attacker, and we next point out that this would be a potentially dangerous option, regardless of the choice of g.

Suppose $C_1, C_2, \ldots, C_s$ are the first s blocks of an intercepted ciphertext message, encrypted using PCBC+, for which the attacker knows the plaintext P_s (corresponding to C_s). Suppose also that message integrity is protected by appending the fixed terminator block P^* to the end of the message prior to encryption. If we further suppose that the attacker has access to a chosen plaintext encryption oracle (a strong assumption, admittedly), then the attacker submits for encryption a message with first block $S \oplus P^* \oplus g(P_s, C_s)$, where S is the 'predicted' IV to be used by the oracle. The first block of the encrypted message will be $C^* = e_K(P^* \oplus g(P_s, C_s))$. The attacker now knows that the ciphertext message $C_1, C_2, \ldots, C_s, C^*$ will be accepted as genuine by the legitimate decrypter, since it is easy to check that the decryption of C^* will yield P^*.

For the remainder of this paper we therefore assume that the IV S is unknown to any attacker — e.g. as would be the case if it is chosen by the sender and sent in encrypted form with the encrypted message. This will nevertheless enable an attacker to force the decrypter to re-use an IV employed to encrypt an intercepted message, without knowing what the value of the IV is — we (implicitly) assume that this attack model applies in the remainder of this paper.

4 An existential forgery attack on PCBC+

We now describe an attack which requires one known ciphertext message, some partial known plaintext corresponding to this ciphertext, and computation with complexity of the order of $2^{n/2}$ operations, where each operation is very simple (much simpler than an encryption operation). Here, as above, n is used to denote the plaintext/ciphertext block length. We also use $\boxplus$ to denote addition modulo 2^n; similarly, $\boxminus$ denotes subtraction modulo 2^n. We first make a trivial observation, whose proof follows immediately from the definition of PCBC+.

Observation 1 *Suppose an attacker knows $C_1, C_2, \ldots, C_t$, a PCBC+ ciphertext message (where C_i is an n-bit block for every i). Suppose also that the attacker knows two consecutive plaintext blocks corresponding to this message: (P_{s-1}, P_s) say, where $1 < s \leq t$. Then the attacker can compute $d_K(C_s)$ as*

$$d_K(C_s) = P_s \oplus (P_{s-1} \boxplus C_{s-1}).$$

We can now give our main result.

Theorem 1. *Suppose, for some $r \geq 2$, an attacker knows r pairs of blocks:*

$$\{(B_i, D_i) \,:\, 1 \leq i \leq r,\ D_i = d_K(B_i)\},$$

where K is a key used to compute one or more PCBC+ ciphertexts. (Such a set can be obtained using r pairs of consecutive known plaintext blocks — see Observation 1). Suppose also that $C_1, C_2, \ldots, C_t$ is a PCBC+ encrypted version of the message $P_1, P_2, \ldots, P_t$, where the final plaintext block P_t is equal to a fixed pattern P^* used to detect changes in the ciphertext and thereby guarantee message integrity. Further suppose that the attacker knows plaintext block P_s for some s satisfying $1 \leq s < t$.

Then, if the integer sequence (u_1, u_2,..., u_w), $w \geq 1$, $1 \leq u_i \leq r$, satisfies

$$E_i = (E_{i-1} \oplus D_{u_i}) \boxplus B_{u_i},\ \ 1 \leq i \leq w$$

where $E_0 = E_w = P_s \boxplus C_s$, then the PCBC+ decrypted version of the ciphertext

$$C_1, C_2, \ldots, C_s, B_{u_1}, B_{u_2}, \ldots, B_{u_w}, C_{s+1}, C_{s+2}, \ldots, C_t$$

is equal to

$$P_1, P_2, \ldots, P_s, E_1 \boxminus B_{u_1}, E_2 \boxminus B_{u_2}, \ldots, E_w \boxminus B_{u_w}, P_{s+1}, P_{s+2}, \ldots, P_t.$$

That is, the modified message is an existential forgery, since the final recovered plaintext block is P^*.

Proof. By definition, the decryption of $C_1, C_2, \ldots, C_s$ will clearly yield the first s plaintext blocks $P_1, P_2, \ldots, P_s$. Next consider the decryption of B_{u_1}. By definition of PCBC+, the recovered plaintext block will equal

$$d_K(B_{u_1}) \oplus (C_s \boxplus P_s) = D_{u_1} \oplus E_0 = E_1 \boxminus B_{u_1}$$

as required. Working inductively, the decrypted version of B_{u_i} $(1 < i \leq w)$ is equal to

$$d_K(B_{u_i}) \oplus (B_{u_{i-1}} \boxplus (E_{i-1} \boxminus B_{u_{i-1}})) = D_{u_i} \oplus E_{i-1} = E_i \boxminus B_{u_i},$$

again as required. The decrypted version of C_{s+1} is equal to

$$\begin{aligned} d_K(c_{s+1}) \oplus (B_{u_w} \boxplus (E_w \boxminus B_{u_w})) &= d_K(C_{s+1}) \oplus E_w \\ &= d_K(C_{s+1}) \oplus (P_s \boxplus C_s) \\ &= P_{s+1}, \end{aligned}$$

and the result follows immediately. □

Thus to find an existential forgery we simply need to find a sequence of positive integers $(u_1, u_2, \ldots, u_w)$, $w \geq 1$, $1 \leq u_i \leq r$, with the property that:

(i) $E_0 = E_w = P_s \boxplus C_s$, and
(ii) $E_i = (E_{i-1} \oplus D_{u_i}) \boxplus B_{u_i}$, $1 \leq i \leq w$.

Such a sequence can be constructed using the 'standard' Birthday Paradox arguments (see, for example, Section 2.1.5 of [1]). The procedure is as follows.

First choose a positive integer v such that $\lfloor r^v \rfloor = 2^{n/2}$. For example, if $n = 64$ and $r = 4$ then $v = 16$. Then generate all r^v possible sequences $(u_1, u_2, \ldots, u_v)$, $1 \leq u_i \leq r$, and for each such sequence compute $E_0, E_1, \ldots, E_v$ using the equations $E_0 = P_s \boxplus C_s$ and $E_i = (E_{i-1} \oplus D_{u_i}) \boxplus B_{u_i}$, $1 \leq i \leq v$. Sort and store all the E_v values.

Now repeat the same process working 'backwards' from $P_s \boxplus C_s$. That is, generate all r^v possible sequences $(u_1, u_2, \ldots, u_v)$, $1 \leq u_i \leq r$, and for each such sequence compute $F_0, F_1, \ldots, F_v$ using the equations $F_v = P_s \boxplus C_s$ and $F_{i-1} = (F_i \boxminus B_{u_i}) \oplus D_{u_i}$, $1 \leq i \leq v$. If any of the values F_0 equal any of the E_v values then the corresponding two sequences can be concatenated to yield a sequence with the desired properties. Because of the choice of the parameter v, there is a good chance of such a match occurring. The attack is now complete.

5 Remarks on the attack on PCBC+

We now analyse the existential forgery attack on PCBC+ presented in Section 4.

5.1 Effectiveness of the attack

The discussion above of the use of a Birthday Paradox search makes the implicit assumption that the values of E_v and F_0 will be randomly distributed across the range of all possible n-bit values. This is clearly not a completely sound assumption, since there is no rigorous evidence that recursively adding and then ex-oring pairs of values from a fixed (small) set of pairs will result in the desired random distribution. Indeed, in some ways this process will clearly not give a random distribution, since if the least significant values of B and D are the same, and $E' = (E \oplus D) \boxplus B$, then E and E' will have the same least significant bit. The precise effectiveness of the attack thus remains to be determined. However, regardless of the choices of D and B, the function $f(X) = (X \oplus D) \boxplus B$ is a permutation on the set of all n-bit values X. Hence it does seem reasonable to assume that the birthday attack will have a good chance of working.

5.2 Attack complexity

First observe that the known plaintext block P_s used in the attack could be one of those used to deduce a value of $d_K(C)$, as per Observation 1. Hence the minimum number of known plaintext blocks necessary to perform the attack is just three, as long as they are all consecutive (since three consecutive blocks will yield

two pairs of consecutive plaintext blocks). Second note that the computations required to perform the attack involve purely comparisons of bit strings, ex-ors of bit strings, and modulo 2^n additions of bit strings. All these operations can be computed very quickly. The total number of operations is clearly $O(2^{n/2})$.

The attack could be performed using a number of known ciphertext messages created using the same key, as long as a plaintext block is known for each. In this case, the first half of one ciphertext message could be joined to the second half of a different ciphertext message (with some intermediary blocks inserted).

5.3 Other integrity protection measures

The above attack assumes that a final plaintext block P^* is used for integrity protection. However, precisely the same approach would work if either the last r blocks of plaintext were set to a fixed pattern, or the final block (or r blocks) were set equal to the first block (or r blocks) of the message.

If the message length is fixed, e.g. by including a string indicating the message length as the first or last plaintext block, then the described attack apparently fails. However, a slightly more complex version of the attack will still work if the attacker knows two plaintext blocks in a ciphertext message $C_1, C_2, \ldots, C_t$ at a distance of precisely w blocks apart: C_r and C_{r+w} say. After a search of the same complexity as described above, a string of w replacement ciphertext blocks can be inserted into the message to replace blocks $C_{r+1}, C_{r+2}, \ldots, C_{r+w}$, without altering the decryption of any subsequent blocks. (The details are straightforward — the main difference is that E_0 and E_w will be different).

In fact, even if the final plaintext block is a CRC computed as a function of all the previous plaintext blocks, an attack along similar lines is still probably possible. In this case, only strings of ciphertext blocks $E_0, E_1, \ldots, E_v$ and $F_0, F_1, \ldots, F_v$ that do not affect the CRC computations should be considered. This will increase the attack complexity to some extent, but otherwise it seems that everything should still work.

5.4 Generalisations of the attack

Finally note that the entire attack strategy outlined in Section 4 could be generalised to other variants of G-PCBC, i.e. to other choices of the function g. Re-examination of the attack reveals that it will still work in exactly the same way as long as g has the following 'invertibility' property. That is, if, given any n-bit blocks X and Y, it is possible to find an n-bit block Z such that $X = g(Y, Z)$, then precisely the same attack strategy will work.

To see this, we outline how to modify the arguments in section 4 to this new scenario. First let g^{-1} be defined such that $g^{-1}(g(X, Y), Y) = X$ for any n-bit blocks X and Y. This 'inverse function' exists by our assumption immediately above. Observation 1 generalises trivially, yielding

$$d_K(C_s) = P_S \oplus g(P_{i-1}, C_{i-1}).$$

Next, as in Theorem 1, suppose $(u_1, u_2, \ldots, u_w)$, $w \geq 1$, $1 \leq u_i \leq r$, satisfies

$$E_i = g(E_{i-1} \oplus D_{u_i}, B_{u_i}), \ 1 \leq i \leq w$$

where $E_0 = E_w = g(P_s, C_s)$. Note that this means that

$$E_{i-1} \oplus D_{u_i} = g^{-1}(E_i, B_{u_i}).$$

Then the G-PCBC decrypted version of the ciphertext message

$$C_1, C_2, \ldots, C_s, B_{u_1}, B_{u_2}, \ldots, B_{u_w}, C_{s+1}, C_{s+2}, \ldots, C_t$$

is equal to

$$P_1, P_2, \ldots, P_s, g^{-1}(E_1, B_{u_1}), g^{-1}(E_2, B_{u_2}), \ldots, g^{-1}(E_w, B_{u_w}), P_{s+1}, P_{s+2}, \ldots, P_t.$$

This follows since:

- the decryption of B_{u_1} yields $d_K(B_{u_1}) \oplus g(P_s, C_s) = D_{u_1} \oplus E_0 = g^{-1}(E_1, B_{u_1})$,
- the decryption of B_{u_i}, $(i > 1)$ yields $d_K(B_{u_i}) \oplus g(B_{u_{i-1}}, g^{-1}(E_{i-1}, B_{u_{i-1}})) = D_{u_i} \oplus E_{i-1} = g^{-1}(E_i, B_{u_i})$, and
- the decrypted version of C_{s+1} equals $d_K(c_{s+1}) \oplus g(B_{u_w}, g^{-1}(E_w, B_{u_w})) = d_K(C_{s+1}) \oplus E_w = d_K(C_{s+1}) \oplus g(P_s, C_s) = P_{s+1}$.

These observations suggest that it is probably dangerous to use any variant of G-PCBC for message integrity, almost regardless of how redundancy is added to the message prior to encryption.

6 M-PCBC

PCBC is appealingly simple to implement, and this motivated recent work by Sierra et al. [13], who define a PCBC variant they call M-PCBC (for *Memory* PCBC). Like G-PCBC and PCBC, M-PCBC requires the message to be divided into a sequence $P_1, P_2, \ldots, P_t$ of n-bit blocks prior to encryption. M-PCBC uses a pair of Initialisation Vectors, which we denote by S_W and S_P, and also a series of intermediate values $W_0, W_1, \ldots, W_t$. Encryption then operates as follows:

$$C_i = e_K(P_i \oplus W_i \oplus P_{i-1}), \ 1 \leq i \leq t,$$

where $W_1 = S_W$, $P_0 = S_P$ and $W_i = G(W_{i-1}, C_{i-1})$, $(1 < i \leq t)$.

The function G maps a pair of n-bit blocks to a single n-bit block, and is defined as follows[1]. Suppose $W = W_L || W_R$ and $C = C_L || C_R$, where $||$ denotes concatenation and W_L, W_R, C_L and C_R are blocks of bits of length $n/2$ (we suppose that n is even, as is always the case in practice). Then

$$G(W, C) = (W_L \oplus W_R) || (C_L \oplus C_R).$$

[1] In order to permit the simplest presentation of the scheme, the notation of [13] has been revised slightly; in the notation of [13], $IV = S_P \oplus S_W$, $IV_2 = S_W$ and $R_i = W_i$.

Hence decryption operates as follows:

$$P_i = d_K(C_i) \oplus W_i \oplus P_{i-1},\ \ 1 \leq i \leq t.$$

To use M-PCBC to protect data integrity, Sierra et al. [13] suggest using the same method as proposed for G-PCBC and PCBC, i.e., they propose adding a fixed final plaintext block prior to encryption. Below, we analyse the effectiveness of M-PCBC for integrity protection on the assumption that this approach is used.

7 Some elementary properties of M-PCBC

We first give an alternative expression for M-PCBC decryption.

Lemma 1. *If $P_1, P_2, \ldots, P_t$ are obtained from the M-PCBC decryption of ciphertext $C_1, C_2, \ldots, C_t$ using key K and Initialisation Vectors S_P and S_W then, if i satisfies $1 \leq i \leq t$:*

$$P_i = S_P \oplus \bigoplus_{j=1}^{i} (d_K(C_j) \oplus W_j).$$

where $W_1 = S_W$, $W_i = G(W_{i-1}, C_{i-1})$, $(1 < i \leq t)$, and we denote the leftmost $n/2$ bits of C_k and W_k by C_k^L and W_k^L respectively, and the rightmost $n/2$ bits of C_k and W_k by C_k^R and W_k^R.

Proof. We prove the result by induction on i.

For the case $i = 1$, observe that $P_1 = d_K(C_1) \oplus W_1 \oplus P_0$, as required. Now suppose that the result holds for $i = r$ (for some r satisfying $1 \leq r < t$). Then:

$$\begin{aligned} P_{r+1} &= d_K(C_{r+1}) \oplus W_{r+1} \oplus P_r \\ &= d_K(C_{r+1}) \oplus W_{r+1} \oplus S_P \oplus \bigoplus_{j=1}^{r} (d_K(C_j) \oplus W_j) \\ &\qquad \text{(by the inductive hypothesis)} \\ &= S_P \oplus \bigoplus_{j=1}^{r+1} (d_K(C_j) \oplus W_j), \end{aligned}$$

as required. The result now follows. □

Lemma 2. *Using the notation and assumptions of Lemma 1,*

$$W_i = (S_W^L \oplus S_W^R \oplus \bigoplus_{k=1}^{i-2} (C_k^L \oplus C_k^R)) || (C_{i-1}^L \oplus C_{i-1}^R),\ \ (1 < i \leq t),$$

where S_W^L and S_W^R are the leftmost and rightmost $n/2$ bits of S_W, respectively. That is, for $1 < i \leq t$:

$$W_i^L = S_W^L \oplus S_W^R \oplus \bigoplus_{k=1}^{i-2} (C_k^L \oplus C_k^R)$$

and

$$W_i^R = C_{i-1}^L \oplus C_{i-1}^R.$$

Proof. We prove the result by induction on i. If $i = 2$, by definition of G and since $W_1 = S_W$, we have $W_2 = G(W_1, C_2) = (S_W^L \oplus S_W^R)||(C_2^L \oplus C_2^R)$, as required. If the result holds for $i = r$ (for some r satisfying $2 \leq r < t$), then:

$$\begin{aligned} W_{r+1} &= G(W_r, C_r) \ \text{(by definition)}, \\ &= (W_r^L \oplus W_r^R)||(C_r^L \oplus C_r^R) \ \text{(by definition of G)}, \\ &= (S_W^L \oplus S_W^R \oplus (\bigoplus_{k=1}^{r-2}(C_k^L \oplus C_k^R)) \oplus (C_{r-1}^L \oplus C_{r-1}^R))||(C_r^L \oplus C_r^R) \end{aligned}$$

(by the inductive hypothesis), and the result now follows. □

These lemmas then enable us to establish the following result, in which the plaintext recovered from an encrypted message can be expressed as a function only of the ciphertext blocks, the Initialisation Vectors, and the secret key K.

Theorem 2. *If $P_1, P_2, \ldots, P_t$ are obtained from the M-PCBC decryption of $C_1, C_2, \ldots, C_t$ using key K and Initialisation Vectors S_W and S_P then:*

$$\begin{aligned} P_1 &= S_P \oplus S_W \oplus d_K(C_1), \\ P_i &= S_P \oplus S_W \oplus \bigoplus_{j=1}^{i} d_K(C_j) \oplus \\ &\qquad (S_W^L \oplus S_W^R \oplus \bigoplus_{\substack{k=1 \\ k \ even}}^{i-2} (C_k^L \oplus C_k^R))||(\bigoplus_{j=1}^{i-1}(C_{j-1}^L \oplus C_{j-1}^R)), \\ &\qquad (i \ even, \ 2 < i \leq t), \\ P_i &= S_P \oplus S_W \oplus \bigoplus_{j=1}^{i} d_K(C_j) \oplus ((\bigoplus_{\substack{k=1 \\ k \ odd}}^{i-2} (C_k^L \oplus C_k^R))||(\bigoplus_{j=1}^{i-1}(C_{j-1}^L \oplus C_{j-1}^R))), \\ &\qquad (i \ odd, \ 2 < i \leq t). \end{aligned}$$

where C_k^L and C_k^R are as defined previously.

Proof. The equation for P_1 follows immediately from Lemma 1. Now suppose i satisfies $2 \leq i \leq t$. Then the result follows from substituting the equation for W_i from Lemma 2 into the equation from Lemma 1. □

8 Breaking M-PCBC

It follows from Theorem 2 that the recovered plaintext P_i is a function only of the set of ciphertext blocks $\{C_1, C_2, \ldots, C_i\}$ (and the key and IVs) and not of the order in which the ciphertext blocks occur (except with respect to whether

the ciphertext blocks appear in an even or an odd position and the values of C_i and C_{i-1}). This enables trivial ciphertext-only 'forgeries' to be constructed, i.e. manipulations of valid messages for which the decrypted version of the final block will remain unchanged (and hence the integrity check will succeed).

For example, in a five-block encrypted message, interchanging the first and third ciphertext blocks will not affect the decryption of the fifth block, although, of course, the first four blocks will be corrupted. This is directly analogous to the simple attacks on PCBC mode due to Kohl [15].

9 Conclusions

The PCBC+ and M-PCBC modes have been described and various properties of each mode exhibited. These properties imply that both modes are unacceptably weak for one of their main intended uses, namely the protection of data integrity. The M-PCBC mode is particularly weak, in that a simple known ciphertext based forgery attack exists, which is easy to perform regardless of the block cipher block length n. The use of one of the recently designed authenticated encryption modes for which a proof of security exists, such as OCB, CCM or EAX, is recommended instead of either of these modes.

Acknowledgements

The author would like to acknowledge helpful comments and advice provided by Lars Knudsen and Caroline Kudla.

References

1. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1997)
2. Bellare, M., Kohno, T., Namprempre, C.: Breaking and provably repairing the SSH authenticated encryption scheme: A case study of the encode-then-encrypt-and-MAC paradigm. ACM Transactions on Information and System Security **7** (2004) 206–241
3. Black, J., Urtubia, H.: Side-channel attacks on symmetric encryption schemes: The case for authenticated encryption. In: Proceedings of the 11th USENIX Security Symposium, San Francisco, CA, USA, August 5-9, 2002, USENIX (2002) 327–338
4. Canvel, B., Hiltgen, A., Vaudenay, S., Vuagnoux, M.: Password interception in a SSL/TLS channel. In Boneh, D., ed.: Advances in Cryptology — CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings. Volume 2729 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2003) 583–599
5. Paterson, K.G., Yau, A.: Padding oracle attacks on the ISO CBC mode padding standard. In Okamoto, T., ed.: Topics in Cryptology — CT-RSA 2004, The Cryptographers' Track at the RSA Conference 2004, San Francisco, CA, USA, February 23-27, 2004, Proceedings. Volume 2964 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2004) 305–323

6. Vaudenay, S.: Security flaws induced by CBC padding — Applications to SSL, IPSEC, WTLS . . . . In Knudsen, L., ed.: Advances in Cryptology — EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 – May 2, 2002, Proceedings. Volume 2332 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2002) 534–545
7. Rogaway, P., Bellare, M., Black, J.: OCB: A block-cipher mode of operation for efficient authenticated encryption. ACM Transactions on Information and System Security **6** (2003) 365–403
8. Bellare, M., Rogaway, P., Wagner, D.: The EAX mode of operation. In Roy, B., Meier, W., eds.: Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers. Volume 3017 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2004) 389–407
9. National Institute of Standards and Technology (NIST): NIST Special Publication 800-38C, Draft Recommendation for Block Cipher Modes of Operation: The CCM Mode For Authentication and Confidentiality. (2003)
10. Whiting, D., Housley, R., Ferguson, N.: RFC 3610, Counter with CBC-MAC (CCM). Internet Engineering Task Force. (2003)
11. International Organization for Standardization Genève, Switzerland: ISO/IEC WD 19772: 2004, Information technology — Security techniques — Authenticated encryption mechanisms. (2004)
12. Dent, A.W., Mitchell, C.J.: User's Guide to Cryptography and Standards. Artech House (2005)
13. Sierra, J.M., Hernandez, J.C., Jayaram, N., Ribagorda, A.: Low computational cost integrity for block ciphers. Future Generation Computer Systems **20** (2004) 857–863
14. Meyer, C.H., Matyas, S.M.: Cryptography: A new dimension in computer data security. John Wiley and Sons, New York (1982)
15. Kohl, J.T.: The use of encryption in Kerberos for network authentication. In Brassard, G., ed.: Advances in Cryptology — CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20–24, 1989, Proceedings. Volume 435 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (1990) 35–43
16. Steiner, J., Neuman, C., Schiller, J.: Kerberos: an authentication service for open network systems. In: Proceedings: Usenix Association, Winter Conference, Dallas 1988, USENIX Association, Berkeley, California (1988) 191–202
17. Gligor, V.G., Donescu, P.: Integrity-aware PCBC encryption schemes. In: Security Protocols, 7th International Workshop, Cambridge, UK, April 19-21, 1999, Proceedings. Volume 1796 of Lecture Notes in Computer Science., Springer-Verlag, Berlin (2000) 153–171
18. Ferguson, N., Whiting, D., Kelsey, J., Wagner, D.: Critical weaknesses of iaPCBC. (1999)